TODO:

Refactoring

-
- Fix movement of entities other than player
- Set dungeon ID
- World entities class to store:
    - Players
    - Static Entities
    - Collectable Entities
    - Characters
- Move methods from world (eg getEntityAtPosition, getEntityResponses) into new WorldEntitiesClass
- Finalise world
- Finalise controller
- Removing entities from the “board” after:
    - They have died (merc, spider, zombie)
    - They have been collected (and add to the inventory instead)
    - They have been destroyed (spawner, anything adjacent to a bomb)
- Comprehensive tests
- What if spider spawn is on top of a boulder
- Final code review call
- Add in one ring use after battle if the player has died (is this the only time it can happen)

TODO

- Inventory - refactor for less coupling
    - Deal with hard mode invincibility
- Randomness - spawning
    - Spider spawn
    - Zombie toast
-
- Save game stuff :(((((((
    - Must implement save game json in each class
-
- Changing strategies when the player is invincible/invincible
- When bombs are dropped they become placed bombs
- UPDATE UML!!!
- Why does battle not end when player health is 0
    - Do we deal with deleting characters?

DONE

- Spider change to static entities
- Spider spawn (in world)
- Battle - armour/one ring spawn? - spawn after battle end
- Redo portal stuff :(

- Changed all entity layers to reflect assumptions
- Finalise starter variable assumptions (amount values etc)
- Drop rate
    - Battle - after a battle armour drops
        - Armour
        - Sword?
        - One ring
- Mercenary interact method
- Ensure types match spec
-

Links to Assignment Materials

SPEC
https://gitlab.cse.unsw.edu.au/COMP2511/21T3/project-specification
REPO
https://gitlab.cse.unsw.edu.au/COMP2511/21T3/groups/W13B_AVOCADO/project
UML
To access:

1. Click the link: https://drive.google.com/file/d/1Wd0djZdC-1Tyk15hMWeh38DdXpE9kTkP/view?usp=sharing
2. At the top of the page, click on the “Open with” drop down.
3. Select diagrams.net to open the draw.io file

Meeting 1 (Lab) - 13/10/21

Initial Thoughts

- Character is observed by mercenaries, spiders, zombies etc.
- Strategy - moving entity has behaviours
- Player class - store collectible items, buildable items as inventory
- Could use states for battles
- Strategy pattern - game mode
- Any assumption that doesn’t overrule the spec is ok
- Mercenary - use shortest path algo to track character (dfs or bfs)
- Collectable entities - consumable, craftables
- Complete the goals of the puzzle to win the game.

Initial assumptions

- Can’t change game mode during the game
- Different “types” of the same puzzle
- Select puzzle at start of game
- Same puzzle type can have different goals
- Restriction on the spawn behaviour of spider - can’t spawn on edges of board

- What happens to the position of collectable items once they have been picked up by the character?
- Health potions “can only be consumed once” - we understand this as meaning that once a health potion has been consumed, it is “discarded”, the character is able to pick up additional health potions later on and consume them
- Can you recollect a bomb after you have placed it?
- Spider and mercenary spawns are pre-placed
- Can you only have 1 armour? Once you have armour you can’t lose it (unless you die)
- Assume toaster breaks with one attack
- What is considered a tick? Movement, action. If the player consumes an item, can they still move on the same tick
- Portals as pairs (A to B and B back to A)
- Assume the outer walls are the boundaries of the map

Category Table

Category	Entities
Collectable	Treasure Key Health potion Invincibility potion Invisibility potion Wood Arrows Bomb Sword Armour The one ring
Movement (interface?)	Spider Zombie Mercenary Player Boulder
Character	Spider Zombie Mercenary Player
Static (don’t move)	Wall Exit Floor switch Door Portal

	Zombie toast spawner Treasure Key Health potion Invincibility potion Invisibility potion Wood Arrows Sword Armour The one ring
Placeable	Bomb
Health Point	Zombie Toaster Spawner Spider Zombie Mercenary Player Sword Invincibility Potion Invisibility Potion Health Potion Bow
Consumable	

List of Entities

- Character (Player)
    - Can move cardinally adjacent (up, down, left, right)
    - Has health and attack damage
- Static entities
    - Wall
    - Exit
    - Boulder
    - Floor switch
    - Door
    - Portal
    - Zombie toast spawner
- Moving entities
    - Spider
    - Zombie
    - Mercenary
- Collectable entities
    - Treasure
    - Key

    - Health potion
    - Invincibility potion
    - Invisibility potion
    - Wood
    - Arrows
    - Bomb
    - Sword
    - Armour
- Rare collectable
    - The one ring
- Buildable entities
    - Bow
    - Shield

Meeting 2 - 17/10/21

- Unique ID for each game
- Open to extension, closed to modification
- Character composes position - position can't exist alone
- Buildable (class??) will look in inventory and check if enough items/resources exist to craft each buildable entity
- What is the frequency of winning the one ring?
- Bow is a weapon
- Have a controller and a world, 1 world for 1 game
- Ally mercenaries can aid the player if they are in the battle radius even if there is an obstacle between them
- Movement could also be implemented with the strategy pattern
- When a player collects an object, it is removed from the worlds objects and added to the characters inventory

Pattern (Initial Ideas)

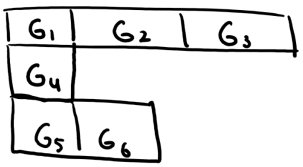
Pattern	Component	Comments
State	Game mode	Once the game mode is chosen at the beginning of the game it can't be changed. The state determines how each object is created
Observer	Observers: zombie toast, mercenary Observed: player	The observers need to know the position or state of the observed to find the shortest path to the observed

State	In battle/ normal gameplay	• For characters as movement is not permitted during battle. • Behaviours such as the mercenary gaining more movement speed when the player is in battle
State	Allegiance (Ally/ Enemy)	Whether the character is an ally or enemy of the player • Affects how the character may interact, for example how they may move (Mercenary moves faster when the player is in battle) • Field in Mercenary class
Strategy	Armour	“Body armour which provides defence and halves enemy attack” Depending on whether the user has armour, the amount of defence/attack changes

https://hackernoon.com/building-a-quest-system-cf7f1d3da132

Goal idea

List < List < Goal >> goals



≈ (G1 OR G2 OR G3)
AND
G4
AND
(G5 OR G6)

Assumptions

- Game play/settings
    - Can't change game mode during the game
    - Each puzzle can be played at each of the gamemodes (peaceful, standard, hard)
    - Each puzzle has initial entity placement and goals set a corresponding JSON files
    - Games are only saved if the "save game" command is run, so if the game is terminated before saving then it isn't saved
    - A tick is defined as any movement or action, which includes
        - Moving positions
        - Consuming an item (potions)
        - Building a buildable item
    - You can only save the current game you are playing
    - Width/height isn't given, so a default spawn radius of x=5, y=5 is set. This is increased as entities are added to the board if their x/y coord is larger
- GameMode
    - Peaceful
        - Enemies don't attack player = mercenary doesn't move
    - Standard
        - Standard HP = 30
    - Hard
        - Hard HP = 15
- Moving Entity
    - Spider
        - Restriction on the spawn behaviour of spider - can't spawn on edges of board
        - Can walk through walls and doors
        - Define the maximum number of spiders as: 6
        - If the spider needs to reverse their direction on a tick (because they are obstructed by a boulder), then they will remain in the same position for that tick, and only start travelling in the new direction on the next tick.
        - Spider does not run away from player when it is invincible
        - HP = 3
        - Attack = 1
        - Spawn rate (or spawn strategy) = 20
    - Zombie Toast
        - Zombie Toast will spawn on a cardinally adjacent cell, provided that there are no entities that zombies cannot step on
            - If there are no valid positions, no zombie will spawn until the next spawn rate
        - HP = 6
        - Attack = 2
        - Armour drop rate = 20%
    - Mercenary

        - Amount of gold (treasure) required to bribe: 2
        - HP = 9
        - Attack = 3
        - Spawn rate (or spawn strategy) = 25
        - Battle radius = 5
        - Armour drop rate = 40%
        - Assume that “within 2 cardinal tiles: as a 2 tile moving distance
    - Player
        - There can only be one player
        - HP - see gamemode above
        - Attack = 3
        - A player is able to attack without a weapon with the above attack stat
            - However, they cannot destroy a zombie toast spawner (as they require a weapon)
    - For all:
        - Only two moving entities can be on one cell at any time
        - This implies that the character can only battle one enemy at the time
        - Once a battle starts, it cannot end until one of the characters is defeated. To continue going through ticks, the player should just just click any direction to continue onto the next attack in battle
        - Given:
            - Character Health = Character Health - ((Enemy Health * Enemy Attack Damage) / 10)
            - Enemy Health = Enemy Health - ((Character Health * Character Attack Damage) / 5)
            - It is assumed that health refers to the CURRENT health of the character/enemy (rather than their initial health), i.e. they get weaker as they are defeated.
- Static Entities
    - Wall
        - The walls on the edge of the map cannot be destroyed (by the bomb)
    - Exit
        - Only triggers the exit win condition when player interacts with it
        - Player only wins the puzzle if other conditions are met, otherwise nothing happens
    - Boulder
        - Can only be pushed by the player
        - Cannot be pushed into the static entities: wall, door, spawner. CAN be pushed onto switch, exit, portal (which teleports it)
        - Cannot be pushed onto enemies
        - Cannot be pushed onto collectable entities
    - Floor Switch
        - Is triggered if a boulder is on it, and untriggered if the boulder is pushed off
        - Only triggered by boulder, otherwise acts as a standard ground cell, i.e. characters can just walk over it
    - Door
        - Has a matching “keyColour” to a key which unique

        - Acts like a wall to enemies even when open (referring to reference implementation).
    - Portal
        - Portals exist as pairs (A to B and B back to A)
        - This means that each map must have 2 (no less or no more) of each portal colour

    - Zombie Toast Spawner
        - Can only be interacted with when the ‘interact’ command in the controller is called, otherwise is used like a wall
        - It breaks on one interaction/attack
        - Spawn rate is determined by current gamemode
- Collectable Entities
    - Treasure
        - Is equivalent to “gold” required to bribe the mercenary
    - Key
        - Has a matching “keyColour” to a door which unique
    - Health potion
        - Restores HP by 10
    - Invincibility potion
        - Spiders don’t run away from players with invincibility potion, as their movement pattern is set
        - Lasts for 10 ticks
    - Invisibility potion
        - Lasts for 10 ticks
        - Player cannot be invincible at the same time as being invisible
    - Wood
    - Arrows
    - Bomb
        - We cannot pick these up again
        - Blast radius = 1
    - Sword
        - Durability: 10
        - Attack power: 3
    - Armour
        - Durability: 7
    - For all
        - Once the character has collected an item, the item is removed from the list of game entities and added to the characters inventory
        - Health potions “can only be consumed once” - we understand this as meaning that once a health potion has been consumed, it is “discarded”, the character is able to pick up additional health potions later on and consume them
    -
- Rare Collectable Entities
    - The One Ring: has a 10% spawn rate
- Buildable Entities
    - Bow

    - Shield
- Other
    - Inventory has unlimited space (no item limit)
    - Weapon and shield stats are stacked
-
-
-
- If the player ends up in a battle while they are invincible, they will not take any health damage
- Player cannot be invincible at the same time as being invisible

Information for our reference:

- Layers
    - Layer 2
        - All enemies
            - Spider, mercenary, zombie toast
        - Player
    - Layer 1 - default - includes statics that we can interact with
        - Walls, boulders etc
        - spawner
        - Collectable entities (enemies can walk on them)
        - Placed bomb
        - Portal
    - Layer 0 - “can be walked on”
        - Floor switch
        - exit

Timeline

- Testing
- World
- Controller stubs
- Character entities
    - Movement methods

- Static entities
    - Interact methods
- Consumables
    - Crafting
- Battle class’
- goals

Meeting 3 - 18/10/21

Objectives of the meeting:

- Testing - unit and blackbox testing
- Delegation of tasks
- Define stub methods for superclasses and interfaces

Completed tasks

- Folder structure completed
- Stubs for classes and interfaces completed
- Testing planned

Super class/ Interface stub	Completed by/Completing
Goal	Mia
Consumable	Mia
GameMode	Claire
Collectables	Jeffrey
Character	Vincent
Buildable	Jeffrey
Static Entity	Mia
Movement	Vincent
Inventory	Nick
World	
Consumable	Jeffrey

- Consumable, has attribute player in order to interact with player through consume() method in consumable entity class.
    - Useful for changes to player and discarding from inventory.

Test Cases

Blackbox testing - the function stubs that are given

- newGame
- saveGame
- loadGame
- allGames
- tick
- Build

- Interact

Use these to help:

- DungeonResponse
- EntityResponse
- ItemResponse
- Position

Unit testing - testing internal backend components

- Character
    - Movement
    - Position
- Static entity (and all subclasses)
- Goal (and all subclasses)
- Game Mode (and all subclasses)
- World

Meeting 4 - 20/10/21

- Discussion about goal implementation: world will keep track of all the goal “aspects” that are needed. World will observe entities that impact goal progress.
- Not going to implement the allegiance interface as there isn’t much use for it right now. Replaced with a boolean in mercenary as this is the only entity that requires an ally/enemy behaviour.
- How to ensure the entity IDs are unique? - counter in world
- Pass world into “tick” functions of each entity
- Define World fields and methods
- Portals - need to be clear on all 4 sides

TODO:

- Character + movement
    - Vincent
    - Mia
- Static + Map
    - Claire
- Collectable + Inventory
    - Jeffrey
    - Nick

Have most of the implementation by next meeting on Sun 24th Oct and add to the uml as you work on the implementation.

Meeting 5 -24/10/21

- Tidy up assumptions
- Start world and controller implementation

Inventory TODO:

- Change collectableItems and buildableItems to a HashMap with id as the key
- For the tick in inventory:
    - we return a list of buildables
    - We update position of all entities within inventory (means all collectible and buildable entities need to have an updatePosition method)
    - Needs to call consume for the used item (world needs check if an item has been collected or not each tick)
- Make a method to see what is craftable
- Attack in Bow (return 2 if there is a bow, return 1 if no bow)
- defenseModifier (return 0.5 if there is armour, return 1 if no armour)
- Add id field to CollectibleEntity (and update all constructors)
- Inventory stores player and the world stores inventory
- Bomb:
    - Consume changes the bomb to be collected
    - Detonate returns a list of all positions around the bomb that needs to be removed
- Health
    - Heal returns the amount to be healed
    - Doesnt implement consumables anymore
- Invincibility Potion
    - Tick decreases duration when active
    - Needs active field
    - Consume activates
- Invisibility Potion
    - Tick decreases duration when active
    - Needs active field
    - Consume activates
- Key
    - Need to know how door key relationship is implemented
        - I ended up just opening the paired door inside key
    - Consume just removes it from inventory
- OneRing
    - Consume just removes from inventory
- Sword
    - Attackmodifier returns its attack power
    - Has durability
    - Consume reduces durability
- Treasure

    - Consume removes
- Wood
    - Consume removes
- Bow
    - Consume reduces durability
- Fix constructors

- Should inventory tick if an invalid itemUsedId is provided?
- What is our blast radius assumption?

Meeting 6 -28/10/21

- Save game
    - JSON for each class
- Save JSON in a certain order (so that anything that relies on another entity is saved in an order that allows us to reload
- CreateSaveJSON - abstract method
- New constructor for classes with extra params/attributes

JSON format:

- World attributes
    - Game mode
    - goal
- Inventory
- Entities
    - Player
    - Other moving
    - Static
    - Collectables
- Battle
- Goals

Pattern (check for final)

Pattern	Component	Comments
State	Game mode	Once the game mode is chosen at the beginning of the game it can't be changed. The state determines how each object is created

Composite	Goals	There may be one goal or composition of goals. There will be a standard interface to check if these are all complete
Strategy	Movement	There are different movement strategies moving entities may follow based on what is happening in the world. Strategies include: ● CircleMovement (spider) ● FollowPlayer ● RandomMovement ● RunAway
Observer	Allies (Mercenaries in range of Player)	Mercenaries observe the player and the player notifies them when they are in range
	TO CHECK BELOW	
Observer	Observers: zombie toast, mercenary Observed: player	The observers need to know the position or state of the observed to find the shortest path to the observed
State	In battle/ normal gameplay	● For characters as movement is not permitted during battle. ● Behaviours such as the mercenary gaining more movement speed when the player is in battle
Strategy	Armour	“Body armour which provides defence and halves enemy attack” Depending on whether the user has armour, the amount of defence/attack changes

Meeting 7 - 29/10/21

- Need to do more comprehensive testing/improve previous tests
- Test on front end

Meeting 8 - 31/10/21

- Consolidation and code review
- Tests writing and front-end visual testing
- Java doc, UML and assumption update
- Coverage checking and final edits

Meeting 9 - 3/11/21

- Milestone 3 meeting
- Keeping track of world states
    - Maybe store a queue(array??) of saved JSONS
    - Store this in world
- Make a boss abstract class
- Logic switches - make all these new statics and just change trigger code?

Meeting 10 - 6/11/21

- Factory pattern for create and load game
- Another passive type and observer pattern for sceptre?
- Ideas on how to implement anduril:
    - State pattern in battle
    - Subclass of battle “boss battle”
    - Boss attack damage modifier in inv
    - Battle class - new strat obj, pass to inv. Inv calls that. Method in inv for “get anduril multiplier”
    - Battle - have current strategy (normal, boss), battleStrategy.attackModifier, passes in inventory. Method in inventory to return weapon list
- To change:
    - Make type of placeBomb “placed_bomb”
- TODO
- Mia, Claire, Vincent:
    - Hydra
    - Assassin
    - Swamp tile
- Jeffrey, Nick:
    - Sun stone
    - Sceptre
    - Midnight armour

Meeting 11 - 8/11/21

- Working on new moving entities + swamp
- Make swamp a state
    - Normal state or in swamp state
    - All moving entities and boulder have to have this state
    - If normal state, move
    - If swamp state, stay for movement factor number of ticks
- Bosses - user decorators - Vincent
- Start looking at extension (light bulb) - Claire
- Start looking into frontend customisations - Mia

Meeting 12 - 10/11/21

- Change gamemode to lowercase and resubmit
- Method forward bounds from factory to world for mercenary bfs

Meeting 13 - 10/11/21

- Standup meeting to see where we are on milestone 3 progress
- Finished logic
- All:
    - Coverage up for tests

    - Fixing code up
    - load/save game fixes
    - UML changes
    - Autotest fixed
- TODO:
    - More testing
    - UML change
    - Javadoc
    - Update Assumptions
    - Update taskboard

Meeting 14 - 14/11/21

- Due to the provided prims algorithm, a default size that's even causes a double layer of walls to be constructed. As the assumed default size is 50 by 50, there is a double layer of wall present at the 48th and 49th row and column.
- Code review, fixing bugs
- ALL:
    - Javadocs
    - Coverage improvements
    - Assumptions update
    - UML changes
    - Fixed Assassin interact issues
- Prims algorithm random world generate finished and tested
- Customisation for frontend